AD-P000 112

EXTRACTING COMPACT OBJECTS USING LINKED PYRAMIDS

T. H. Hong
M. Shneier

Computer Vision Laboratory, Computer Science Center
University of Maryland, College Park, MD 20742

ABSTRACT

Compact objects of arbitrary size are extracted from images using a combination of three-pyramid-based representations of image features. A gray-scale linked pyramid is used to smooth the image into uniform regions. A "surroundedness" pyramid is used to identify regions of interest, and a linked edge pyramid is used to delimit the boundaries of the compact objects.

1. Introduction

Many image processing tasks require the extraction of objects from a background. Most notable among these is target detection. In many cases there is some a priori knowledge about the shapes and sizes of the objects, which could aid in their extraction. Unfortunately, it has not normally been possible to extract objects that have the right size and shape without extracting other, unwanted objects as well. Removing the unwanted objects then requires another stage of processing, which can be very complicated if the desired objects are embedded in background clutter.

This paper presents a pyramid-based method of extractin compact objects that is able to apply knowled about the size and shape of an object directly to the segmentation process, to avoid extracting unwanted regions. The method provides solutions to a group of problems, including object detection, edge completion, and region filling. It makes use of both gray-scale and edge information. In addition, it computes a surroundedness measure for each pixel, representing the degree to which that pixel is locally surrounded by edges. All three sets of information - gray level, edge magnitude and direction, and surroundedness - are represented in pyramid structures, and it is the interaction between the different types of information at each level of each pyramid that leads to the final segmentation. The representations are, themselves, built on one another. A gray-level pyramid is used to construct an edge pyramid, which is in turn used to construct a surroundedness pyramid.

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76-C-0138 (DARPA Order 3206) is gratefully adknowledged, as is the help of Clara Robertson in preparing this paper.

The gray-scale pyramid is used to segment the original image into smooth regions that are not necessarily connected (Section 2.1). It is common for an object to belong entirely to one of these regions, but the algorithm does not require this to be the case. The edge pyramid (Section 2.2) is used in two ways. The edges indicate parts of the image that could be individual objects, enabling the objects to be separated from the regions extracted by the gray-level pyramid. The edges also serve as the basis for constructing the pyramid of surroundedness scores (Section 2.3).

The surroundedness scores are used to find starting points for a combined region-growing and region-splitting process. The growth of the region is controlled by the gray-level pyramid, and the region is pruned by the edge pyramid. In this sense, the method is analogous to the "superslice" algorithm (Milgram, 1979) and to the relaxation method of Danker and Rosenfeld (1979). One of the notable features of the method is that the region does not "leak" through holes in its border. This is partly because of the pyramid's tendency to bridge small gaps as the resolution decreases from level to level.

A previous use of a pyramid process for extracting compact objects (Shneier, 1979) made use only of gray values and a compactness measure. For each compact region that was discovered, a threshold was computed and applied in a square region of the original image to extract the object. The current method does not use a threshold to extract the regions, but makes use of edge information to determine the shapes and sizes of the regions.

The process of constructing the pyramids is described in Section 2, and the succeeding section describes how each pyramid is used to arrive at the final result. Examples are given of applying the system to a set of images, and the results are compared with those obtained in a recent segmentation study (Hartley et al., 1981).

In the following sections, the pixels at each level of the various pyramids play two roles. They are points in an image at some level of a pyramid, and are also nodes in the tree structure defined by the links between levels in the pyramids. Both names will be used interchangeably.

2. Constructing the pyramids

2.1 Gray-scale pyramid

A gray-scale pyramid is a sequence of square images, each a lower-resolution version of its predecessor. The kind of pyramid used in this work is the linked structure defined by Burt et al., (1980). It is constructed as follows.

Each level is formed by summarizing a 4 by 4 neighborhood in the preceding level. The neighborhoods are overlapped fifty percent vertically and horizontally so that each pixel has four "fathers" at the next level, and sixteen "sons" at the previous level. The average or the median of the sixteen sons can be used as the summarizing value for their father. In the implementation, the average value was used.

The entire pyramid is constructed in this way, up to the level at which there are only four pixels. There follows an iterated linking process in which each node is linked to that one of its four fathers whose gray value is most similar to its own. A father can thus have up to sixteen sons, while a son can have only one father. After the links have been established, each node recomputes its gray value based only on the values of the sons linked to it. This process is iterated, and usually stabilizes after a few iterations.

At this stage, each pixel at the bottom level of the pyramid (the original image) is linked through some sequence of ancestors to one of the four pixels in the topmost (2 by 2) level of the pyramid. Each topmost node thus represents some region in the original image, which can be extracted by following links down the pyramid. If the values of pixels in the original image are replaced by the corresponding values of their ancestors, a segmentation of the image into at most four regions is obtained. It is not necessary that these regions be connected.

The segmentation defined by this procedure is not necessarily in terms of objects and background. Indeed, for the image in Figure 1, the chromosomes are extracted as one component, while the background is segmented into three components of slightly different average gray-value. Often, a desired region belongs to one of the four components, but is lost among the other parts of the image that link to the same component. As an example, notice that one of the small chromosomes in Figure 1b disappears entirely. The procedure defined in this paper is largely concerned with isolating individual parts of the four components into separate objects, although it is also able to merge parts from different components into a single object. The process relies on edge and surroundedness information to find the subcomponents to be extracted.

Figure 1 shows an image and the results of iterating the gray-level linking process. The resulting preliminary segmentation forms the imput to the rest of the procedure.

2.2 Edge Pyramid

The edge pyramid is constructed by first building a gray-level pyramid and then applying an edge operator at each level to produce an edge pyramid (Hong et al., 1981). The gray-level pyramid used for extracting edges was based on non-overlapped 2 by 2 blocks, and the values at each level were defined as the medians rather than the means of the values in the blocks at the level below. This reduces the amount of blurring and distortion of the edges (Tanimoto, 1976).

The edge operator that was used is one that scored highest in the edge evaluation tests of Kitchen and Rosenfeld (1981). It is the three-level template operator (Abdou and Pratt, 1979) which uses eight direction masks, e.g.

$$\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \quad \text{and} \quad \begin{matrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{matrix}$$

The edge detection is followed by a non-maximum suppression stage. A 3 by 3 window is placed around each edge point. The direction of the edge is used to find the two edge points to use for non-maximum suppression. If the edge point has a magnitude greater than both points, and a direction difference of less than 45 degrees, it survives; otherwise, it is deleted. Figure 2a shows an edge pyramid constructed from the chromosome image.

Edges, too, are linked together between levels. For linking purposes, the pyramid is assumed to map each point to a 4 by 4 region in the level below. Once again, each son has four potential fathers and each father has sixteen sons. Linking proceeds bottom-up. Each son compares his direction with those of his four fathers, and chooses the father whose direction is most compatible. If the difference in directions is less than some threshold (here 46 degrees), the son is linked to the father. Otherwise, the son becomes the root of a tree. Ties are broken by choosing the first father that satisfies the criteria. The direction of a son is updated to become the average of the son's direction and the father's direction, but the process is not iterated.

2.3 Surroundedness pyramid

The edges at each level of the edge pyramid are directed in such a way that the brighter side of the edge is to its right. This information could be used by itself to prune the gray-level pyramid by demanding that the gray levels at positions corresponding to opposite sides of an edge obey this constraint. Such a process would not necessarily lead to a segmentation into compact objects. It is first necessary to identify the edges that bound compact objects, and to ignore all other edges. A procedure for finding such edges was described in Hong et al. (1981).

In the current system, however, the aim is to extract the interiors of compact regions. The process is applied at each level of the pyramid, and compact objects of different sizes are identified

at different levels. There are two stages involved in finding eompact regions from edge information.

First, the skeleton of a region is found by looking at 5 by 5 neighborhoods of each point. There is no need to look further than two points on either side of a pixel, because, if there are no edges within this distance, the object will become more compact at the next higher level of the pyramid, where the process is applied as well. The aim is to find interior points of a region that are surrounded by edge points with compatible directions.

Let x be the central point in a 5 by 5 neighborhood (Figure 3). The remaining points in the neighborhood are divided into three classes. The points marked A are the immediate neighbors of x, while those marked B and C are more distant from x. The numbers associated with each point are their chain code orientations in units of 45 degrees. Finding the skeleton proceeds as follows.

If the edge magnitude of x is not zero, ignore this point, because x is not interior.

If the magnitude is zero, check the neighbors of x:

1. For each type A neighbor of x whose edge magnitude is not zero, the edge direction of A is allowed to differ from its chain-code direction by no more than some threshold (here 23 degrees). For example, the edge direction of the point immediately East of x must lie between -23 degrees and +23 degrees, while the edge direction of the point North-East of x must lie between 23 degrees and 45 degrees. That is, the edge directions should be consistent with the edges of a closed region. If this condition is met, the score for the particular direetion from x is set to 1. The score is a measure of how central the point is, i.e., of its membership in the skeleton of the region. For each point, there are eight slots for scores, corresponding to eight directions. A perfect border around x would result in all eight slots being set to 1. Note that more than one point in the 5 by 5 neighborhood can set the same slot value.

2. If the magnitude of a type A neighbor of x is zero, the neighboring type B point is examined as above. If its edge direction is compatible with its grid position, the score for x is set to 1.

3. For all type C points whose edge magnitude is not zero, the corresponding direction slot for x is set to 1 if the direction of the point is within 23 degrees of the chain-code position. For type C points, however, the chain-code direction is calculated at 45 * chain-code number + 23, because type C points are offset an extra 23 degrees from x.

Notice that all the type A and type C points contribute to the score for x, but type B points only contribute if the neighboring type A point is not an edge point. This is because closer edges are assumed to block the effects of edges that are more distant, and hence less likely to belong to the same object. This is particularly important at high levels of the pyramid where the objects are very close together.

When the scoring process has been applied to each 5 by 5 neighborhood at each level in the pyramid, the second stage of finding compact regions is performed. The purpose of the second stage is to propagate the score of the skeleton out to the borders of the region. For the seeond stage, the score is computed as the sum of the slot values. A threshold is applied to decide what score values are considered to constitute valid skeleton points (here a seore of 5 out of a possible 8 was used). For each such point the following procedure is performed.

1. For all type A or C points whose edge magnitudes are not zero and whose edge directions are eompatible (as in the previous step), assign a new score which is the maximum of the current score and the sum of the slot values for x (the skeleton point).

2. For type A points whose edge magnitude is zero, check the corresponding type B point. If its magnitude is not zero and its direction is eompatible, assign a new score to both the type A point and the type B point. In each case the score is the maximum of the score for x and the current score for the point.

When both steps of the process have been completed, each compact region will contain a set of high scores, as will the edge points surrounding the region (Figure 2b). These points define the extent of the region at the particular level in the pyramid. To extract the corresponding region in the original image requires the use of both the gray-level and the edge pyramids. The particular scoring function used does not have any special significance, and it is likely that other functions would perform equally well.

Note that no thresholding was used to discard edges with very low magnitudes. It is sometimes useful to keep only the strong edges, and so avoid extracting objects with very low contrast that are invisible to the human eye. To a large extent the loss of resolution at higher levels of the pyramid achieves this automatically, but it is true that at low levels in the pyramid a lot of small noise regions might be extracted. Examples of the improved performance resulting from thresholding the edge magnitudes are shown in Section 4.

The surroundedness pyramid has no links between the levels. As a result, compact objects can be detected at more than one level of the pyramid. In previous work (Hong et al., 1981) links were established, and the object was detected at the highest level at which it was well defined. Such

a process would probably work for the current pyramid structure as well.

3. Extracting the compact regions

The most obvious way of extracting compact regions from a given level in the surroundedness pyramid is first to find all points that have a high surroundedness score. These points can then be projected down to the base level by finding the corresponding points in the gray-level pyramid and following their links. Unfortunately, this simple process results in regions that are displaced, misshapen, and which have holes and protrusions that do not appear in the original objects.

There are a number of reasons for these imperfections, analysis of which leads to a more complex extraction process, but one that produces regions that are much closer to the actual shapes of the objects. The flaws can arise from a poor initial segmentation in the gray-level pyramid and from displaced or missing edges in the edge pyramid. Poor edge data also lead to incorrect surroundedness information, and this also must be improved.

The compact region developed from the edge information can be incorrect for two reasons. First, the edges could be misplaced due to the averaging in the pyramid process and the non-maximum suppression applied at each level. Second, there may be missing or noisy edges. To correct the placement of the points, use is made of information from the gray-level pyramid. If the object is known to have a particular color, then all points with that color that are in the compact region (i.e. have a high surroundedness score) can be called object points, and the rest can be ignored. Alternatively, if the object is known to be, say, the brightest region in the image, the gray value of the brightest node of the 2 by 2 level in the gray-level pyramid can be projected down and intersected with the compact points to give a more accurate compact region. Usually, however, the relative brightness of the object is not known, or the object may have more than one color, so that a more conservative approach has to be taken. This involves a local process to identify the set of gray values that occur most commonly in the interior of the compact region. These values are taken as representing the object, and neighboring points with the same gray values are added to the starting set to give a new compact region whose position is more accurate because it is derived both from edge- and region-based properties.

To correct for missing edges, the structure of the edge pyramid is used. As the resolution of the pyramid increases towards its base, the positions of the edges become more and more accurate, but the gaps become larger and larger. By fitting lines through existing edge points in a top-down process, the gaps can be filled in relatively cheaply, and should approximate the actual contours of the boundary more and more closely as the resolution increases.

Another problem that arises from using edge information is that holes can appear inside object regions because of noise in the image. For most applications that were implemented, no edge magnitude thresholding was performed, and for all applications, no thresholding was performed above the base level of the pyramid. As a result, edge points with very low magnitude often appear in the interior of objects. Again, by taking advantage of the pyramid process, it is possible to remove these edges and so ensure that holes do not appear inside the objects. A characteristic of noisy edges is that they do not survive as the resolution of the image is reduced at successive pyramid levels. By examining the sons of interior points and deleting those that are edge points, the interior of the region can be cleaned up. Of course, it is possible for holes that are real features to be eliminated in this way, and it is likely that edges with magnitudes above some threshold should be retained.

The final difficulty of using naive projection to find the compact regions is that the objects that are found have misshapen boundaries. In some places, the boundary might extend into the background, while in others it might not extend out to the actual border. It is even possible for the simple projection process to give rise to disjoint regions at the base of the pyramid. This problem is overcome by projecting down the gray values level by level, and using the edges at successive levels to delimit the borders.

The process of extracting regions involves a simultaneous addition and deletion of nodes in the gray-level pyramid, guided by the edge and surroundedness pyramids. Nodes are added if they are on the interior side of an edge and adjacent to a compact point. They are deleted if they are on the outside of an edge belonging to the compact object. The additions and deletions are performed top-down at each level of the pyramid below the level at which the compact object was discovered. The result is a region whose outline closely follows the edge bounding the object, and which is tolerant of gaps in the edge information. This is similar to the process described by Strong and Rosenfeld (1973), but occurs vertically across levels of the pyramid, instead of horizontally within a level. In more detail, the process is as follows.

1. Project the gray values from the top (2 by 2) level of the pyramid down to the level at which the compact object was discovered (i.e., the level at which it received an above-threshold surroundedness score). Call this level L.

2. Choose points to be considered as part of the object from among the points belonging to the compact region as follows. For every point x in level L that is an interior point (i.e., has a high score and is not an edge point), examine the surrounding 5 by 5 window. If x has the same value as the majority of its neighbors, then x is considered a valid object point. This ensures that points that have gray values that belong to the background, or have a mixture of the region and background colors, are not included.

3. Expand the set of points belonging to the compact object by again looking at 5 by 5 neighborhoods, this time for all points x at level L, regardless of whether they are interior points or not. If x has neighbors in the 5 by 5 region that were chosen as object points in the previous step, then x is marked as an object point if x has the same gray value as one of those points. This compensates for shifts in the edge positions due to the pyramid process and the non-maximum suppression.

4. Project the nodes in the enlarged compact region down one level in the gray-level pyramid, to level L-1.

5. Examine interior points of the compact region in the edge pyramid at level L. If any of the central four sons of an interior point are edge points, delete them. This cleans out noisy edge points in the interior of the object at level L-1.

6. At level L-1, expand the compact region by examining edge points that link all the way to level L. If these edge points have interior neighbors that are not part of the region, add them in regardless of their gray value. This expands the region to fit the boundary at the current level.

7. Fit lines through the edge points of 7 by 7 neighborhoods at level L-1 (see below). Delete points that lie outside these lines if they are part of the compact region. This ensures that the region does not grow outside the edge boundary, and prevents leaks where no edges exist.

8. Repeat steps 4 - 7 for levels L-2, L-3,... until the bottom of the pyramid is reached. At this stage, the compact region has been extracted.

Lines are fitted to edge points below level L to fill in gaps in the edges. For every edge point x that links to the border of an object at level L, a set of points (e.g., those marked a in Figure 4 and their rotations) is examined if x satisfies the following conditions.

1. x must not be surrounded by interior points. This assumes that the objects do not have holes in them, and can be relaxed if necessary.

2. There is no edge parallel to x in the area marked by a's in Figure 4. This is because the parallel edge will prune the region and, since edge magnitudes were not used, the outermost edge is considered the real edge at the current level.

If both conditions are satisfied, all the points marked a are pruned. In the implementation, points were only deleted if they did not link to <u>any</u> compact object. This was because all compact regions were being extracted simultaneously, and it was possible for points from a different object to appear in the neighborhood, especially at high levels in the pyramid.

The reason for projecting the values from the 2 by 2 level to level L and using the set of points that have the most common gray values is to alleviate effects that the edge construction process has on the position of edges. Assume that an object is represented mostly by a single gray value in the original image, and that this consistency is preserved at all levels of the pyramid. Then, so long as the edges do not shift too far, the intersection of the compact region and the set of points with the most common gray values is a good seed for growing the region. Adding in points that are immediate neighbors of the seed points and that have the same gray values ensures that the region is shifted appropriately. It does not matter too much if the corresponding region at the bottom of the pyramid is too large, because the pruning that takes place at lower levels will make sure that the region stays within the boundaries defined by the edges. Note that the shift in the edges is greatest at the top of the pyramid, and becomes less and less as the base level (the original image) is approached. Because of the links between levels, the shifting is not particularly important. Every projection follows the links, both in the gray-level and the edge pyramids, so that the size and position of the region converges to the true size and position of the corresponding object as the base of the pyramid is approached.

Note that no threshold was applied to the edge magnitudes, so that many weak edges remain at each level. Most of these do not form links to the next level, or, at least do not survive as the size of the region that contains them shrinks. The noise-cleaning step examines interior (i.e. non-edge) points at one level and deletes any of their central 2 by 2 sons that are edge points. This step can sometimes cause interior detail to be lost. For example, in the image of Figure 5, the central dark region is filled in. Usually, however, the process ensures that there are not holes in the final object.

The step of expanding the region to conform with the edge data accounts both for the fact that the gray-level pyramid might not match the edge pyramid exactly and for the possibility that the gray values of the object might not be uniform. Many objects exhibit a smooth transition with the background. By expanding the region, guided by the edges, it is possible to account for variations in gray values.

Region splitting is applied for similar reasons. If there is no change in gray values between the object and the background in the gray-level pyramid, then many points outside the object will be linked to nodes that are interior nodes at a higher level in the pyramid. When the gray values are so similar, it often happens that no edges are found at the corresponding positions on the edge pyramid. By the nature of the pyramid, however, a missing segment becomes smaller and smaller as the

height of the pyramid increases. By interpolating across small breaks at each level, a close approximation to the actual boundary can be obtained. This interpolation is done by fitting lines through the edges at each level. All nodes that lie outside these lines are pruned, while those inside are added to the object. As the resolution increases down the pyramid, the fitting process approximates the object boundary more and more closely.

4. Examples

The procedure was applied to a set of FLIR images and to a picture of a number of chromosomes of varying sizes. On the whole, the results were very satisfactory, although the method is less successful when the objects are so small as to appear only in the original image. In these cases, there is no smoothing effect from the pyramids and, because there is no thresholding of the edge magnitudes, a number of small noise regions are extracted together with the desired regions.

Figure 6 shows a very clean example of the system's abilities. The original image consists of a number of chromosomes against a dark background. There are no objects so small as to be visible only at the full-resolution level of the pyramid, and the objects are spread out by size across the next two levels. Thus, the smaller chromosomes appear in Figure 6a, while the larger chromosomes appear in Figure 6b. The larger chromosomes are also extracted in Figure 6a because their surroundedness score is high enough at this level. The process mentioned earlier of choosing the best level at which to extract an object would enable the larger chromosomes to be extracted only at the higher level.

It should be realized that each chromosome is extracted individually, even though the gray-level pyramid links them into a single top-level node. The chromosomes are extracted cleanly, despite the gaps in edge information evident in the edge images (Figure 2), and despite the fact that one small chromosome is totally lost in the background of the gray-level pyramid.

Figure 7 shows an example of the expansion of the gray-level pyramid region to fit the edge image. In the upper left image, the original gray-scale tank merges fairly smoothly with the background. This results in an original compact region smaller than the actual tank (bottom left). The compact object was actually found at the 8 by 8 level of the pyramid, and the bottom right image shows the results of adding in points on the inside of the edge data at the 16 by 16, 32 by 32, and 64 by 64 levels of the pyramid. The result is a region whose shape is a close approximation to the shape of the actual object.

Figure 8 shows an example where parts of the region outside the object are discarded by the pruning step. A node was removed because it was on the wrong side the region boundary, resulting in a more accurate outline. In fact, such pruning happens in almost all the images.

Figure 9 shows what happens when the objects being sought are too small. If an object is not large enough to be represented at a level above the original image, the only filtering taking place is due to the surroundedness scoring. It is possible for a single noise point to give rise to a compact region, and this would be detected in addition to any legitimate targets. Noise cleaning at this level eliminates many of the detected objects, but can remove the desired objects as well. By thresholding the edge magnitudes, however, a much better result can be obtained. A similar improvement could be expected if the surroundedness scoring took the edge magnitudes into account. Even without any thresholding, the number of regions detected is still less than that for the gray-level linking based segmentation. On these same images objects that are large enough to survive even to the first level above the original image are detected with almost no background clutter. Figures 10a to 10s show the results obtained when the edge magnitude is thresholded (at 15). Figures 11a to 11i, 12a to 12f, and 13a to 13p show the objects extracted at successively higher levels without edge magnitude thresholding.

In the segmentation study of Hartley et al. (1981), the gray-level linking method of segmentation performed reasonably well, except for the detection of a large number of unwanted objects (false alarms). The current method, being based on the gray-level linking process, is guaranteed to do no worse than that method. In fact, the results show that the method significantly reduces the number of false alarms, and often eliminates them entirely. The method can also be tuned to detect objects of a particular range of sizes, and does so with no extra processing. If the method were to be ranked using the scoring function of Hartley et al., it would rank ahead of all the methods they tested. (Table I). Note that most of the images in the study had to be sampled down to 64 by 64 pixels because that is the largest size the program can handle. For those images for which sampling was not necessary (11-30), the method performed better than the others. Overall, the method was as good at detecting targets as the best method in that study, but had a lower false alarm rate, and no extra detections. The method would probably perform even better if it were re-implemented to handle full-resolution images.

5. Discussion and Conclusions

A method has been presented that extracts compact objects from images. The method uses three kinds of pyramid-based representations. The first is a gray-level pyramid, with links between points at successive levels. The second is a pyramid of edge information for each level, and the third is a surroundedness pyramid that reflects the compactness of regions at each level.

The results of applying the method to a number of images indicate that it is successful in extracting compact objects so long as they are large enough to survive at least to the second level of the pyramid. The extracted objects have borders that closely follow the outlines in the original

scene, as found by the edge detector, and very few extraneous regions are usually detected. Even in the cases in which the objects are very small, they are still usually extracted, although a number of unwanted regions might also be extracted. By thresholding the edge magnitudes of the original image, most of the unwanted regions can be discarded, leaving only the compact objects. It can also be seen that the process extracts only compact regions. For example, the road in Figure 14 is not extracted, because it is elongated rather than compact.

Levine (1980) discussed a pyramid-based algorithm for region analysis that is related to the approach presented in this paper. He made use of three color pyramids, a texture pyramid, and an edge pyramid. None of the pyramids were constructed using overlapping regions, and the edge pyramid was formed by ORing 4 by 4 regions of an original edge image to produce the successive levels. The aim of the research was not to extract objects with particular shapes, but to segment a scene into regions. Processing involved finding points as far away from the borders of regions as possible, by finding the levels in the edge pyramid abovae which a set of edges disappeared. These points then served as seeds for growing regions by projection in the pyramids. At each level, the boundaries between regions were refined by a close examination of the neighboring points. When the final projection was completed, a clean-up process was used to merge small regions with adjacent larger regions. The method proposed in this paper makes more use of local gray values in the analysis, and does not need to perform any postprocessing of the image.

Earlier work has also concerned the problem of filling in regions from broken edge information. Strong and Rosenfeld (1973) describe an iterative procedure that simultaneously grows regions and fills in gaps in the borders. The method described here has advantages in that the speed with which regions can be filled in is significantly greater in the pyramid, as is the distance over which gaps in the edges can be bridged.

Danker and Rosenfeld (1979) examined the use of pyramids to speed up the propagation of edge and region labels in their relaxation scheme for extracting regions, but their results were inconclusive. Given the ability to perform operations in parallel, the current method can be made very efficient. The pyramids are all constructed in one pass, although the gray-value pyramid linking process is iterated. Later processing involves a single pass through the pyramid, starting at the level at which the compact object is found, and ending at the level of the original image. All processing within and across levels is local in nature, so that the potential exists for real-time implementation of the algorithm. To make the results comparable with the study of Hartley et al., the gray-level linking process was iterated. It is not clear that this is necessary because the process does not depend on having regions with uniform colors.

It would be of interest to extend this work by devising scoring functions to detect elongated objects, for example, or objects of arbitrary shape. With a small set of primitive shape recognizers it would be possible to build a powerful system that could selectively extract objects having a wide variety of shapes.

REFERENCES

1. I. E. Abdou and W. K. Pratt, Quantitative design and evaluation of enhancement/thresholding edge detectors. Proceedings IEEE 67, 1979, 753-767.

2. P. Burt, T. H. Hong, and A. Rosenfeld, Segmentation and estimation of image region properties through cooperative hierarchical computation. IEEE Transactions Systems, Man, Cybernetics 11, 1981, 802-809.

3. A. Danker and A. Rosenfeld, Blob detection by relaxation. IEEE Transactions Pattern Analysis Machine Intelligence 3, 1981, 79-92.

4. R. L. Hartley, L. J. Kitchen, C. Y. Wang, and A. Rosenfeld, A comparative study of segmentation algorithms for FLIR images. TR-1104, Computer Science Center, University of Maryland, College Park, MD, September 1981.

5. T. H. Hong, M. Shneier, and A. Rosenfeld, Border extraction using linked edged pyramids. TR-1080, Computer Science Center, University of Maryland, College Park, MD, July 1981.

6. L. J. Kitchen and A. Rosenfeld, Edge evaluation using local edge coherence. IEEE Transactions Systems, Man, Cybernetics 11, 1981, 597-605.

7. M. D. Levine, Region analysis using a pyramid data structure. In Structured Computer Vision (S. Tanimoto and A. Klinger, eds.) Academic Press, New York, 1980, 57-100.

8. D. L. Milgram, Region extraction using convergent evidence. Computer Graphics Image Processing 11, 1979, 1-12.

9. M. Shneier, Using pyramids to define local thresholds for blob detection. TR-808, Computer Science Center, University of Maryland, College Park, MD, September 1979.

10. J. P. Strong III and A. Rosenfeld, A region coloring technique for scene analysis. Comm. ACM 16, 1973, 237-246.

11. S. L. Tanimoto, Pictorial feature distortion in a pyramid. Computer Graphics Image Processing 5, 1976, 330-352.

Images	Targets	Method	Correctly detected	Extra detections	False alarms	Segmentation accuracy
2-10 (Navy, China Lake)	8	2-class relaxation	0	0	43	-
		3-class relaxation	2	0	67	0.70
		Pyramid linking	0	0	145	-
		Superspike	3	0	77	0.51
		Surroundedness pyramid	4	0	32	0.60
11-30 (NVL data)	80		40	0	92	0.73
			20	8	92	0.49
			72	32	392	0.67
			76	24	60	0.64
			76	0	16	0.73
31-36 (Air Force, TASVAL)	6		2	0	9	0.74
			3	1	27	0.73
			3	2	100	0.57
			6	1	63	0.60
			5	0	11	0.70
55-70 (NVL flight test)	32		2	0	6	0.67
			13	1	19	0.65
			4	0	38	0.80
			26	1	2	0.73
			26	0	7	0.60
Overall	126		44	0	150	0.73
			38	10	205	0.58
			79	34	675	0.68
			111	26	202	0.66
			111	0	67	0.69

Table I. Summary of results for the comparative segmentation study.

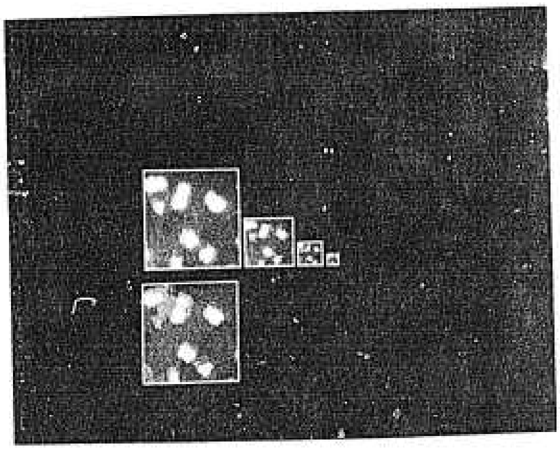


Figure 1. Top: a gray-level pyramid for a chromosome image. Bottom: the results of iterating the gray-level linking process (10 iterations).

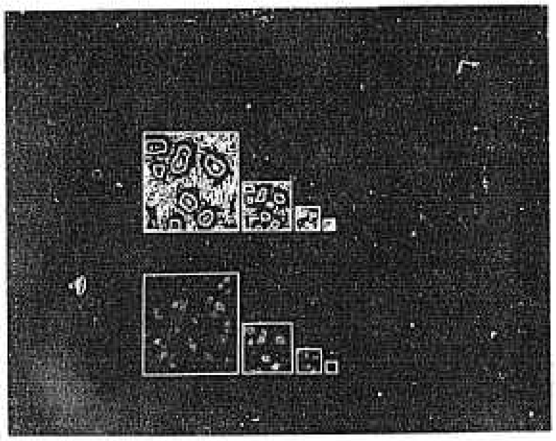


Figure 2. Top: an edge pyramid for the chromosome image. Bottom: a surroundedness pyramid for the chromosome image.

B3	C2	B2	C1	B1
C3	A3	A2	A1	C0
B4	A4	x	A0	B0
C4	A5	A6	A7	C7
B5	C5	B6	C6	B7

Figure 3. The 5 by 5 neighborhood for computing surroundedness scores. The numbers denote chain-code directions.

				a		
				a	a	
				a	a	a
			←x	a	a	a
				a	a	a
				a	a	
				a		

a	a	a	x↗			
a	a	a	a			
a	a	a	a			
a	a	a	a			

Figure 4. The 7 by 7 neighborhoods used to fit lines through edge points. The arrow indicates the direction of the edge point x, and the a's indicate the region that is examined. Rotations of these patterns are used for other edge directions.

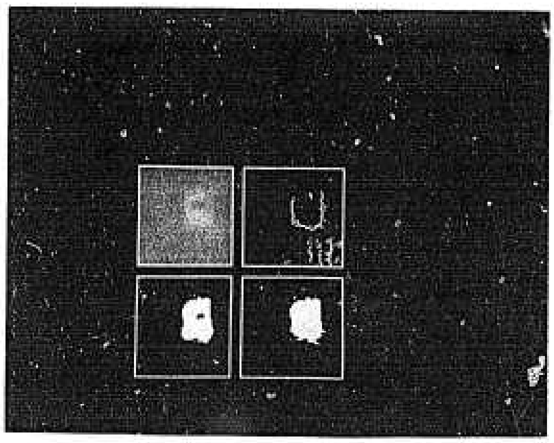
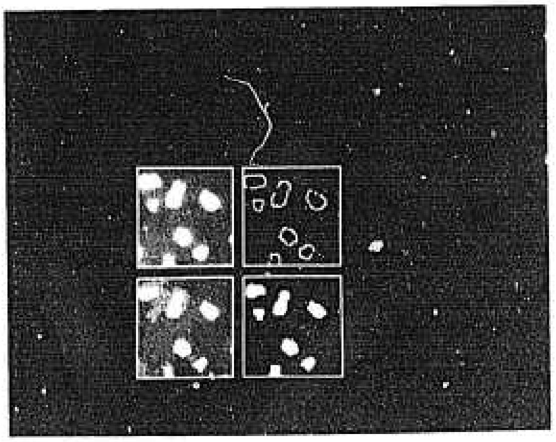


Figure 5. Top left: the original FLIR image of an armored personnel carrier. Top right: the edge image projected down from the level at which the compact object was found (8 by 8). Bottom left: the compact object found at level 3 (8 by 8) without deleting interior edges. Bottom right: the result of applying the whole process to the image. The hole in the middle has been filled in.



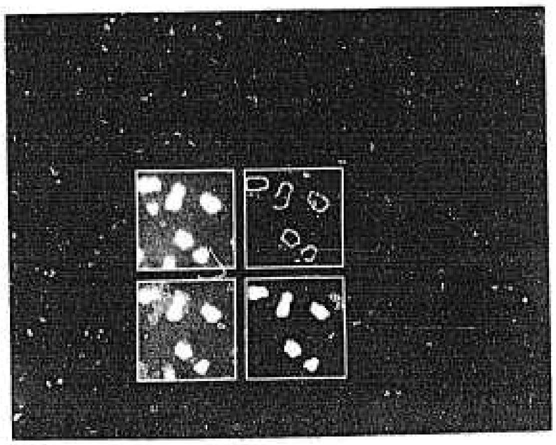


Figure 6. a) The chromosomes extracted at level 1 (32 by 32), the first level above the original image.

b) The chromosomes extracted at level 2 (16 by 16).

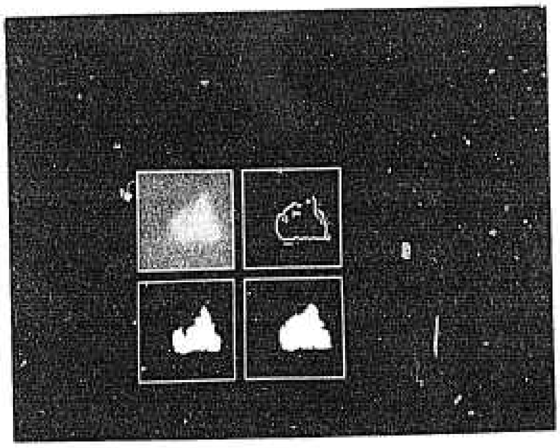


Figure 7. Top left: original FLIR image of a tank. Top right: edge image projected from the 8 by 8 level. Bottom left: the compact object found at the 8 by 8 level. Bottom right: the results of adding points to fit the edge data.

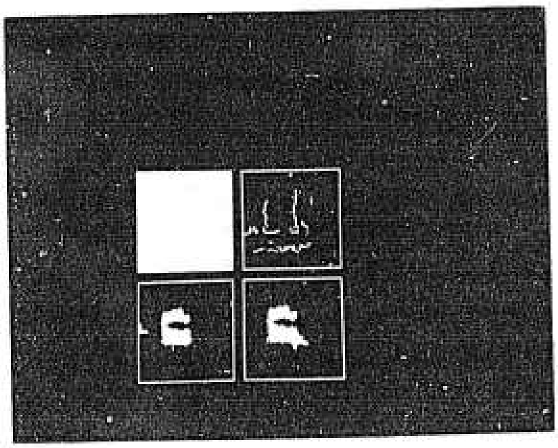


Figure 8. Top left: original FLIR image. Top right: edge image projected from the 8 by 8 level. Bottom left: compact object without pruning. Bottom right: compact object after pruning.

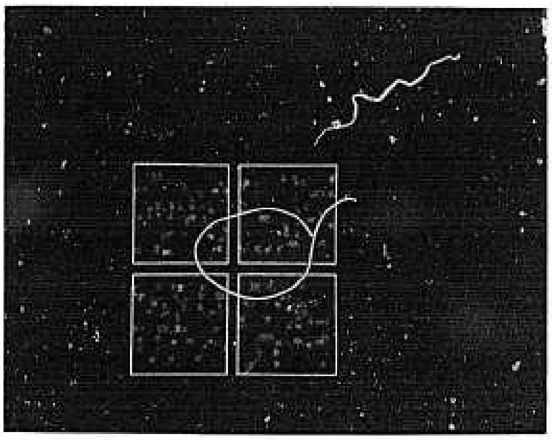


Figure 9. The results of running the process when the objects are found only at the base level of the pyramid (no thresholding).

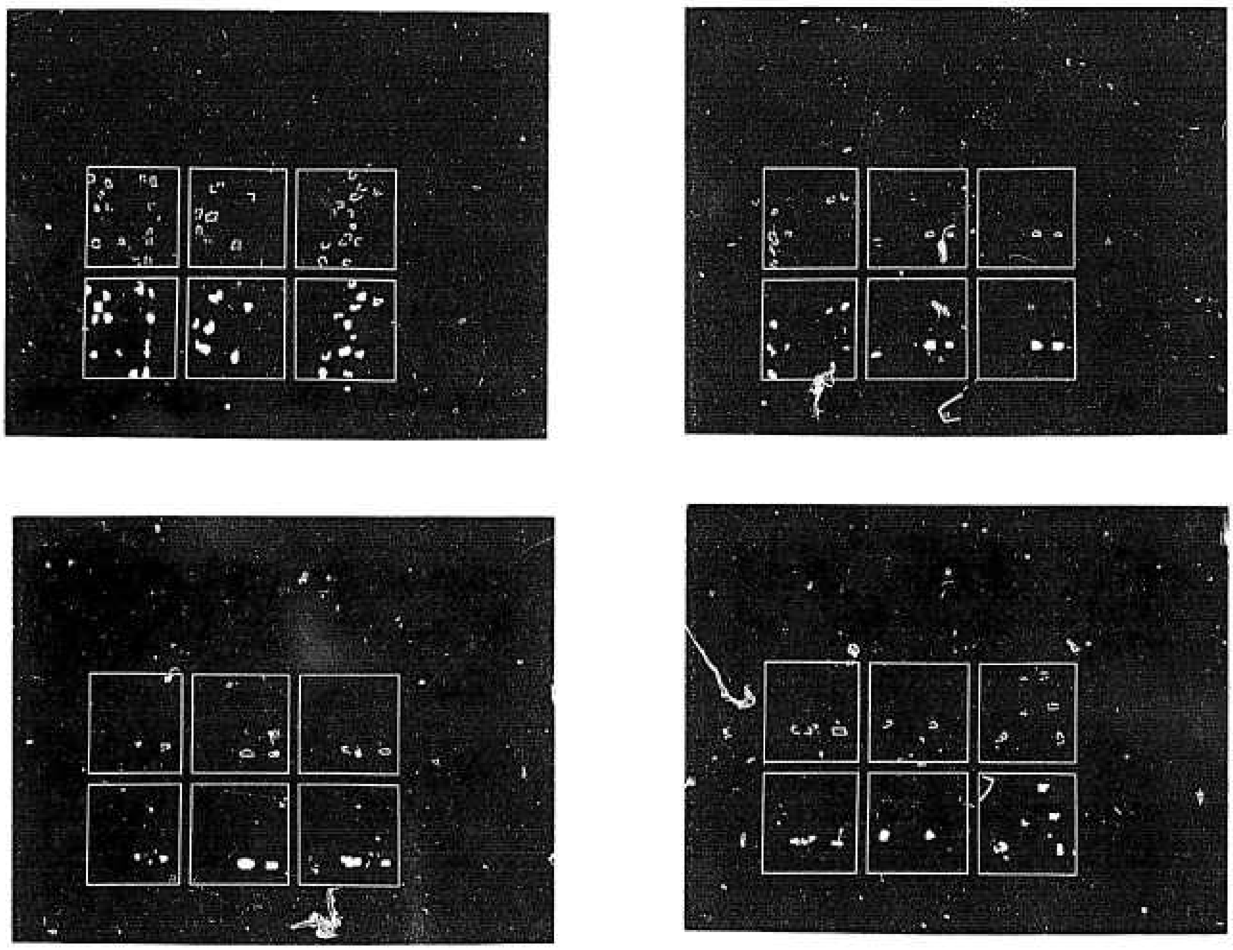
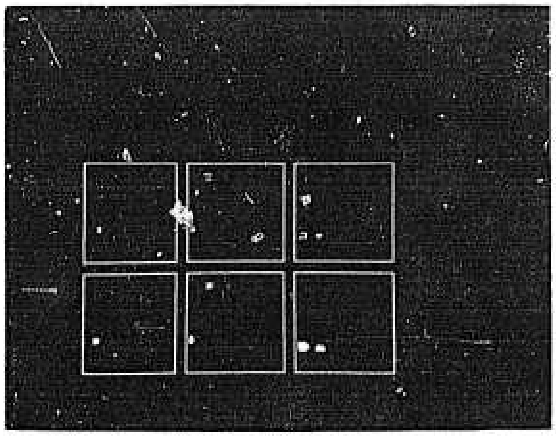
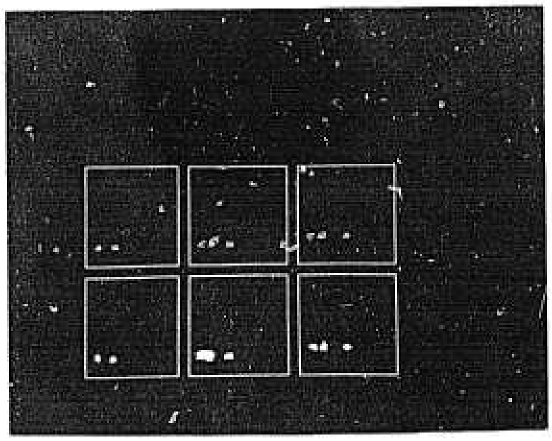


Figure 10. a-s. The results of running the process with the edge magnitude thresholded at 15.

a b c   d e f

g h i   j k ℓ





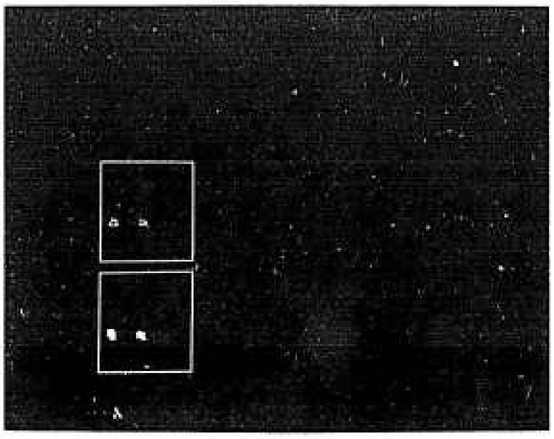


Figure 10, cont'd.:

m n o   p q r
s

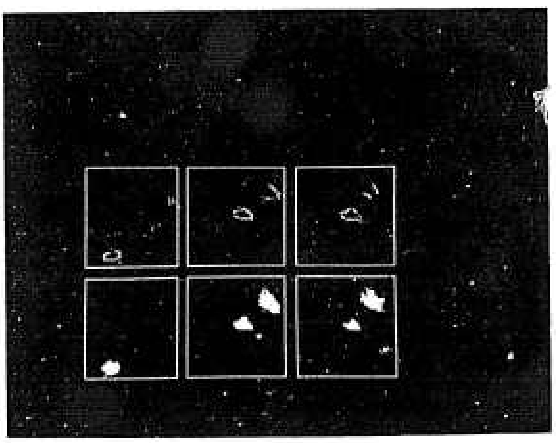
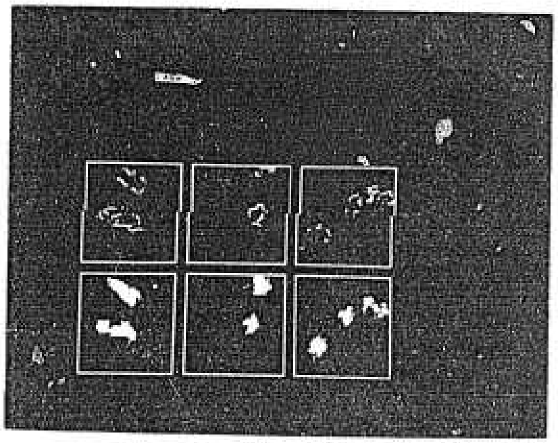


Figure 11. a-i. The results of running the process on images from Figure 9 when the objects are extracted at level 1 (32 by 32).

a b c



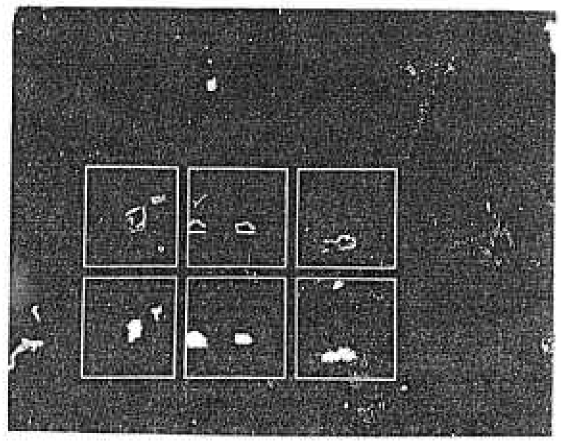
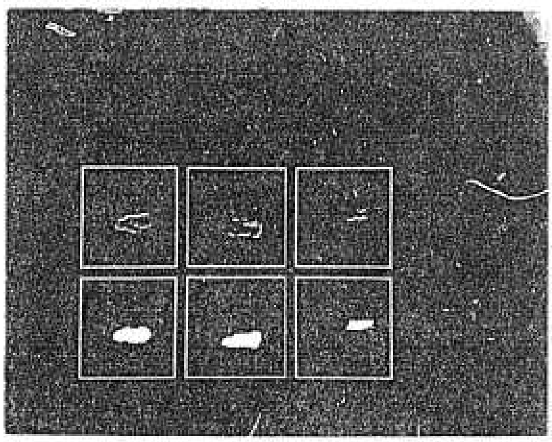
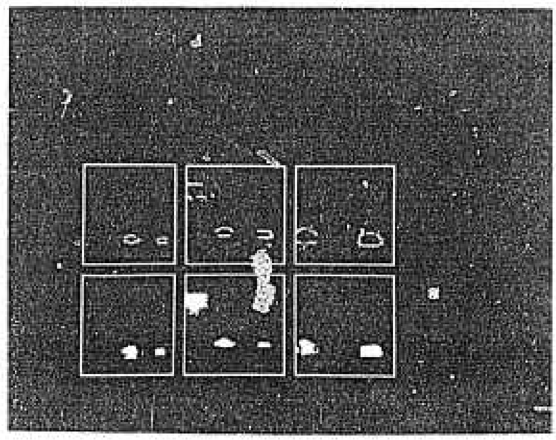


Figure 11, cont'd.: d e f

g h i





a b c

d e f

Figure 12. a-f. The results of running the process on images where the objects are extracted at level 2 (16 by 16).

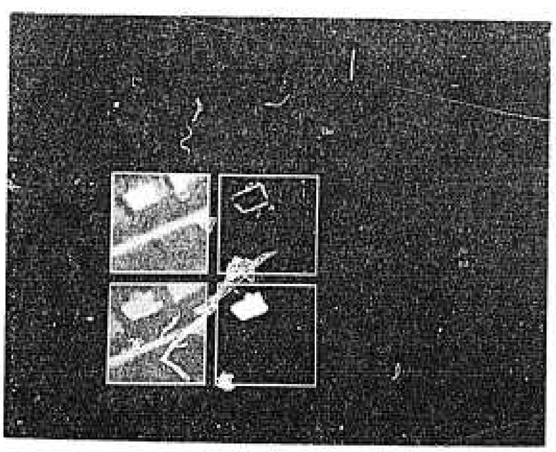


Figure 14. Part of a suburban scene with a road and a house. The house is compact enough to be extracted, but the road is not.

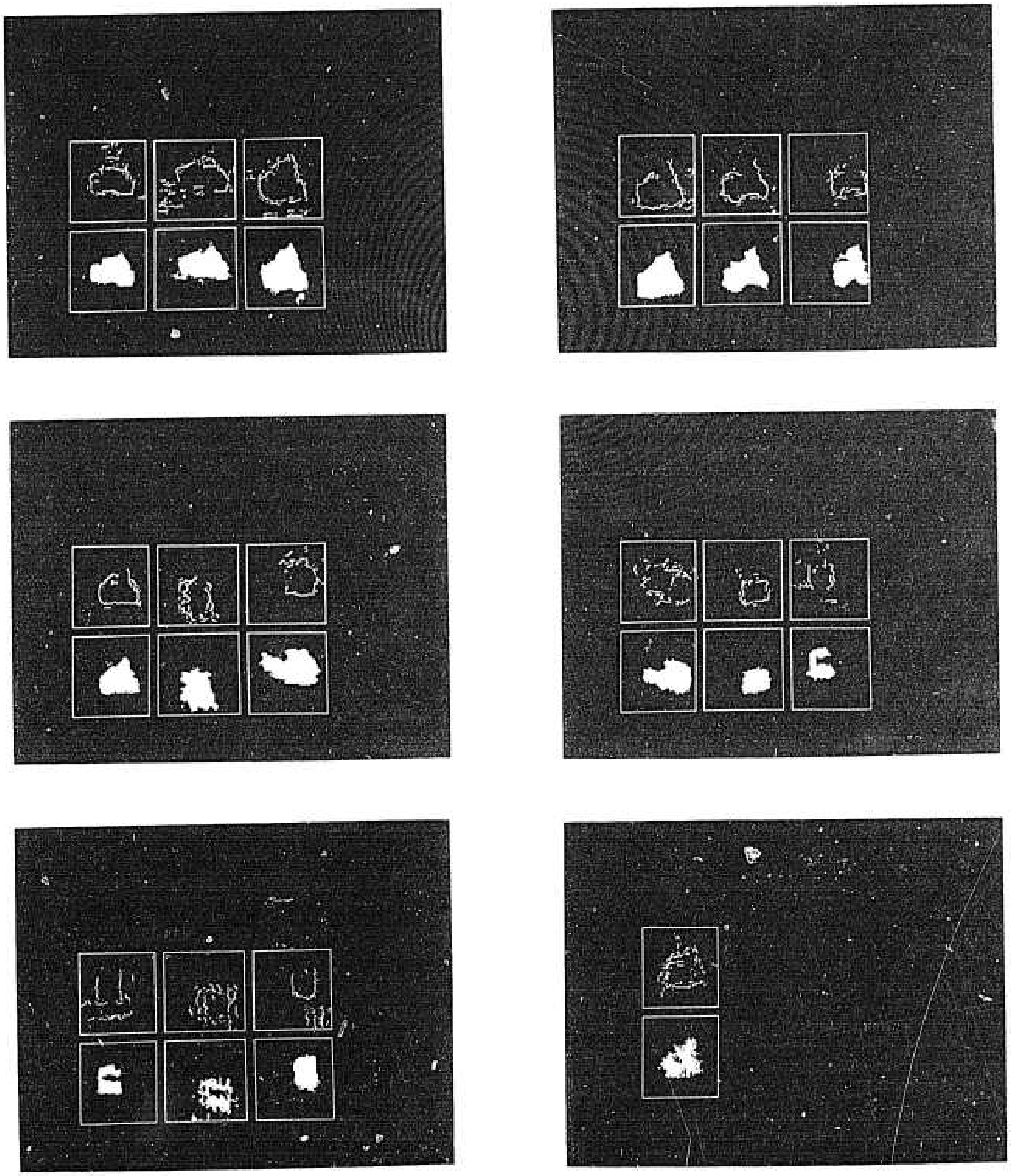


Figure 13. a-p. The results of running the process on images where the objects are extracted at level 3 (8 by 8).

a b c d e f

g h i j k ℓ

m n o p